Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

Trends in Information Systems Curricula: Object-Oriented Topics

Bill Hardgrave
University of Arkansas

David Douglas
University of Arkansas

Trends in Information Systems Curricula: Object-Oriented Topics

Bill C. Hardgrave
David E. Douglas
Computer Information Systems and Quantitative Analysis
University of Arkansas

Introduction

Several significant events during the past few years in the field of object-oriented technology have reshaped the industry. Among these are the rapid increase in the use and interest of Java, and the adoption of the Unified Modeling Language (UML) by the Object Management Group (OMG) in November 1997. These, and other, events continue to change the pattern of OO use in industry and course offerings in academia. This study specifically looks at the changes in academia. As OO continues its growth in industry, institutions of higher education will be expected to prepare their students for the OO development environment. In an effort to evaluate academia's ability to produce OO developers, information has been collected from higher education (IS departments) in 1992, 1996, and most recently, in 1998 regarding OO topic offerings. This paper reports the results of the most recent findings as compared to the 1992 and 1996 data. Specifically, changes in OO programming languages, OO methods, and OO tools are presented. The trends exhibited during this time frame should provide guidance for both academia and industry as to the commitment of IS degree programs in providing a significant share of the talent pool to support the shift to OO development.

Methods and Results

Respondents

An e-mail questionnaire was sent to 435 IS department chairs listed in the *1995 Directory of Management Information Systems Faculty*. Seventy e-mails were returned because of bad addresses, leaving 365 departments that successfully received the questionnaire. One hundred six usable responses were received representing a 29 percent response rate. Details concerning the data collection for the 1992 and 1996 surveys can be found in Massey and Douglas (1993) and Douglas and Hardgrave (1997).

In 1992, 38 percent of the schools were teaching OO topics; in 1996, 71 percent were teaching OO and 17 percent had plans to do so. This study (1998) found 84 percent of the schools teaching OO - remarkably close to the previous (1996) study's prediction of 88 percent (71 percent teaching and 17 percent planning to teach). Overall, an upward trend continues in the number of schools teaching OO topics.

Programming Languages

Table 1 summarizes the OO programming languages (OOPLs) identified from the three surveys as being taught in IS degree programs. In 1992 and 1996, C++ represented the language most often taught. However, in 1998, Visual Basic unseated C++ as the most popular language. Visual Basic also represents the largest increase in the most recent survey, jumping from 8% in 1996 to 70% in 1998. Java also increased from 0% in 1996 to 44% in 1998. The 1996 finding is not surprising given that Java was formally introduced by Sun Microsystems in May 1995 (Deitel and Deitel, 1997). The 1998 finding of 44% for Java is amazing given the infancy of the language. Other increases from 1996 to 1998 include: Object COBOL from 0% to 7%; and C++ from 48% to 53%. Languages that appear to be declining, based upon the most recent data, include: Ada (-7%), Eiffel (-2%), Lisp (-1%), Object Pascal (-5%), Objective-C (-1%); and Smalltalk (-5%).

Interestingly, the "pure" OO languages (Henderson-Sellers and Edwards, 1994), such as Actor, Eiffel, Simula, and Smalltalk are used very little and/or are declining in use (Java is the exception to this trend). "Hybrid" languages (Henderson-Sellers and Edwards, 1994), such as C++ and Object COBOL, continue to increase in popularity. Visual Basic, considered an object-based language (Brown, 1997), is growing rapidly.

The number of different OO languages taught is down from 8 in 1996 to 6 in 1998, perhaps indicating a move toward a 'standard' set of languages. The number of institutions teaching more than one OO language has almost doubled from 28% in 1996 to 55% in 1998. This would imply a wider use of OO languages in the curriculum, possibly supplanting traditional languages, such as COBOL.

Development Methods

The OO development methods currently taught are shown in Table 2. In the most recent survey, UML is the most popular "method". Without getting into an explanation or debate here, UML is clearly not a methodology (Henderson-Sellers, 1998).

Table 1: OO Languages Taught

Language	1992 (n = 92)	1996 (n = 101)	1998 (n = 106)
	% (#)	% (#)	% (#)
Actor	2% (2)	0% (0)	0% (0)
Ada	2% (2)	7% (7)	0% (0)
C++	20% (18)	48% (48)	53% (56)
CLOS	0% (0)	0% (0)	0% (0)
Eiffel	0% (0)	2% (2)	0% (0)
Java	0% (0)	0% (0)	42% (45)
Lisp	1% (1)	2% (2)	1% (1)
Object COBOL	0% (0)	0% (0)	7% (7)
Object Pascal	3% (3)	5% (5)	0% (0)
Objective-C	0% (0)	1% (1)	0% (0)
Simula	0% (0)	0% (0)	0% (0)
Smalltalk	10% (9)	9% (9)	5% (5)
Visual Basic	2% (2)	8% (8)	70% (74)
# of languages	7	8	6
teach more than 1 OO language	7% (6)	28% (28)	55% (58)

However, it is the newly adopted notation set and metamodel which incorporates elements of Booch's method, OMT, and Jacobson's Objectory. Thus, it is not surprising that UML is the most popular, given the popularity of Booch and OMT as indicated in the 1996 survey. It is surprising that schools have made the switch so quickly to UML when the adoption only became official in November 1997 and very little information (in the form of textbooks) is available. Booch, OMT, and Jacobson's Objectory are still being taught, but these should gradually disappear within the next few years as UML becomes more well known. Open, a relatively new and complete methodology, is being used at one respondent's university.

Coad and Yourdon's method continues to increase in usage. Despite the growing popularity of Booch, OMT, and Jacobson, and the ultimate adoption of UML, Coad and Yourdon's method remains strong. This may be due to its simplicity and its strong relationship to traditional development methods. As an introduction to OO analysis and design, Coad and Yourdon's method may be the simplest to teach.

Table 2: OO Development Methodologies

	1992 (n = 92)	1996 (n = 101)	1998 (n = 106)
	% (#)	% (#)	% (#)
Booch	11% (10)	21% (21)	18% (19)
Coad & Yourdon	15% (14)	16% (16)	19% (20)
Jacobson	0% (0)	3% (3)	6% (6)
OMT	0% (0)	10% (10)	14% (15)
Open	–	–	1% (1)
UML	–	–	21% (22)

Tools: OO DBMS and OO CASE

The current use of object-oriented database management systems (OODBMS) and OO CASE tools is low. Of those schools teaching OO technology, only 2% are utilizing OODBMS, and 35% are using OO CASE tools for instructional purposes. Both reflect decreases from the 1996 survey of 15% and 44% for OODBMS and OO CASE tools, respectively (see Table 3).

Table 3: OO Tools

Tools	1996 (n = 72)	1998 (n = 89)
	% (#)	% (#)
OODBMS	15% (11)	2% (2)
OO CASE	44% (32)	35% (31)

For OODBMS, the decrease may be attributed to the lack of standards and a solid theoretical basis (Blaha and Premerlani, 1998; Brown, 1997). The use of OODBMS in industry is also very low (Jones, 1995). Currently, the Object Database Management Group is working to define a set of standards for object-oriented databases (Brown, 1997; Henderson-Sellers and Edwards, 1994). The recent rise in so-called "Object/Relational DBMS" (Hunter, 1997) may also be impacting the use of a "pure" OODBMS. Due to the overwhelming use of existing relational database technology, companies and schools may be looking at Object/Relational databases as their migration path.

The use of OO CASE tools has probably been hampered by the recent adoption of UML. As shown earlier, 21% of the schools are now teaching UML, but the OO CASE tool vendors are behind in providing OO CASE tool support for the new UML standard. OO CASE tools that supported Booch and OMT, for example, may no longer be useful for UML due to the changes in notation. Now

that UML has been adopted, one would expect to see a plethora of OO CASE tools become available and subsequent utilization of those OO CASE tools in higher education curricula.

Course Offerings

Of the 106 respondents, 84% teach at least one OO language, 51% teach at least one OO method, and 47% teach at least one OO language and one OO method. Those schools offering OO courses (i.e., teach OO language and/or OO method courses) provide a median of 2 courses per school (mean = 2.84; mode = 2). Many indicated that OO is integrated in several courses rather than having separate courses for OO. Thus, the actual number of courses incorporating OO would be higher than 2, although we cannot say by how much. For example, Object COBOL may be integrated into an existing COBOL class, or OO methods may be integrated into an existing analysis and design class. The bureaucratic difficulty in introducing new courses into a curriculum may be forcing many programs to introduce OO into existing courses.

Summary

In the six year period of time (1992 - 1998), the percentage of IS departments offering OO technologies has increased from 38 percent to 84 percent which exceeds the percentage of organizations currently developing, and planning to develop OO applications in the next few years (*Computerworld*, 1994).

Visual Basic, C++, and Java are the most popular languages in IS programs. Visual Basic and Java have increased tremendously in the past few years, whereas C++ increased slightly from 1996. Object COBOL, although not yet standardized, exhibited signs of early use (7%). On the other hand, Smalltalk continues to decrease in use (now down to 5%).

UML, recently adopted by OMG as the standard OO notation set, is the most utilized method. Booch, OMT, and Jacobson's Objectory continue to be widely used, but should decrease with the adoption of UML. Coad and Yourdon's method continues to be popular as an OO method to teach.

Disappointingly, but not unexpected, OODBMS and OO CASE tools are not widely used. There are many reasons for the lack of use, not the least of which, is the lack of good tools provided by tool vendors. OODBMS have yet to find their niche in the marketplace, and the lack of a standard notation set has hindered a consistent set of OO CASE tools.

As companies move to OO, they will depend on IS departments, among others, to provide OO developers. Results of this study indicate that IS departments are increasing their offerings of OO technologies and are preparing their graduates for jobs in an IS field moving to OO development. Overall, more OO courses are being taught by more IS departments throughout the United States and Canada compared to just a few years ago.

References

Blaha, M., and Premerlani, W., *Object-Oriented Modeling and Design for Database Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1998.

Brown, D., *Object-Oriented Analysis: Objects in Plain English*, John Wiley & Sons, New York, NY, 1997.

Coad, P., *Object Models: Strategies, Patterns, and Applications*, Yourdon Press / Prentice Hall, 2nd edition, 1997.

"Corporate IS Considers Object Plans," *Computerworld*, 28 (4), Jan 24, 1994, 61, 64.

Deitel, H.M., and Deitel, P.J., *Java: How to Program*, Prentice-Hall, Englewood Cliffs, NJ, 1997.

Douglas, D., and Hardgrave, B.C., "Object-Oriented Trends: Information Systems Degree Programs," *1997 AIS Americas Conference on Information Systems*, August 1997, 754-755.

Henderson-Sellers, B., and Edwards, J., *BookTwo of Object-Oriented Knowledge: The Working Object*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

Hunter, S.K., "Cutting to the Chase: Marketing Wars are in Full Gear, But What Does ORDBMS Really Mean?" *Object Magazine*, 7 (6), Aug 1997, 32-41.

Jones, N., "Results of an Interactive Survey on Object Orientation," Gartner Group, Research Note: T-700-1192, July 12, 1995.

Massey, P.D, and Douglas, D.E., "Object-Oriented Technologies in Information Systems Curricula," *Journal of Computer Information Systems*, 34 (1), Fall 1993, 3-7.